

Ap No: 8

Solve the Crossword Puzzle as constraint satisfaction problem

Srinidhi.V
18C102

Aim:

To solve the crossword puzzle problem as constraint satisfaction problem.

Algorithm:

1. Start

2. Function Arc-Consistent

2.1 let Q be initialized to contain all the arcs of CS-problem.

2.2 Reduce all the domains of CS-problem so that they satisfy the unary constraints

2.3 Remove the arc from Q until Q is empty.

2.3.1 If the domain is empty, return failure.

2.3.2 Else add to Q all arcs and return with success.

3. Function Arc-Reduce

3.1 let $change$ be a boolean variable initialized to false.

3.2 For each value in the domain find a value in the domain of variable such that it satisfies the binary constraints of the problem.

3.3 If there is no such variable, remove value from domain and set $change$ to true.

3.4 Return the value of $change$.

4. Stop.

Sample Input And Output:

18C102

Given list of words:

		1	2	3	4	5
AFT	LASER					
		1	1A	2A		3A
ALE	LEE	2	-	-	-	
EEL	LINE	3	-	4A	5D	
HEEL	SAILS					
		4	6D	-	7A	
HIKE	SHEET					
		5	8A			
HOSES	STEER					
		6	-	-	-	-
KEEL	TIE					
KNOT						

(x_i, x_j) New D_i , if changed Added

$(1A, 2D)$ Hoses, Laser $(3D, 1A)$ $(3D, 8A)$ Sails, Steer $(1A, 3D)$

$(3D, 1A)$ Sails, Sheet, Steer $(4A, 3D)$ $(7A, 3D)$ $(4A, 3D)$ $(7A, 3D)$
 $(8A, 3D)$ $(5D, 8A)$

$(4A, 3D)$ Heel, Hike, Keel $(2D, 4A)$ $(5D, 4A)$ $(6D, 8A)$ Ale

$(7A, 3D)$ Ale, eel, Lee, Tie $(2D, 7A)$ $(5D, 7A)$ $(2D, 4A)$ Sails

$(8A, 3D)$ $(3D, 4A)$ Steer $(8A, 3D)$

$(2D, 4A)$ Sails, Sheet, Steer $(1A, 2D)$ $(7A, 2D)$ $(8A, 2D)$ $(1A, 3D)$ Hoses $(2D, 1A)$

$(5D, 7A)$ Keel $(4A, 5D)$ $(4A, 3D)$

$(1A, 2D)$ $(7A, 3D)$

$(7A, 2D)$ Eel, Lee $(3D, 7A)$ $(5D, 7A)$ $(1A, 2D)$

$(8A, 2D)$ Hoses, Laser $(3D, 8A)$ $(5D, 8A)$ $(6D, 8A)$ $(7A, 2D)$

$(7A, 5D)$ $(8A, 2D)$ $(8A, 3D)$ Laser $(2D, 8A)$ $(5D, 8A)$

$(8A, 5D)$ $(8D, 6A)$

$(4A, 5D)$ Hike $(2D, 4A)$ $(3D, 4A)$ $(2D, 1A)$ $(2D, 8A)$

$(3D, 7A)$ $(5D, 8A)$ $(6D, 8A)$

$(1D, 7A)$

1 A O S E S

2 - - A - T

3 - H I K E

4 A - L E E

5 L A S E R

6 E - - L -

Program:

```
from collections import defaultdict
```

```
# ***** INITIALIZATION FUNCTIONS
***** #
```

```
def initDomains():
    domain, position = dict(), dict()
    li2, li3 = ["HEEL", "HIKE", "KEEL", "KNOT", "LINE"], ["AFT", "ALE", "EEL", "LEE", "TIE"]
    li1 = ["HOSES", "LASER", "SAILS", "SHEET", "STEER"]
    domain = {"1A" : li1[:], "4A" : li2[:], "7A" : li3[:], "8A" : li1[:],
              "2D" : li1[:], "3D" : li1[:], "5D" : li2[:], "6D" : li3[:]}
    position = {"1A" : (0, 0, 'A'), "4A" : (2, 1, 'A'), "7A" : (3, 2, 'A'), "8A" : (4, 0, 'A'),
               "2D" : (0, 2, 'D'), "3D" : (0, 4, 'D'), "5D" : (2, 3, 'D'), "6D" : (3, 0, 'D')}
    return domain, position
```

```
def initConstraints():
    constr = defaultdict(list)
    li = [('1A', '2D', 3, 1), ('1A', '3D', 5, 1), ('4A', '2D', 2, 3), ('4A', '5D', 3, 1),
          ('4A', '3D', 4, 3), ('7A', '2D', 1, 4), ('7A', '5D', 2, 2), ('7A', '3D', 3, 4),
          ('8A', '6D', 1, 2), ('8A', '2D', 3, 5), ('8A', '5D', 4, 3), ('8A', '3D', 5, 5)]
    for var1, var2, x, y in li:
        constr[var1].append((var2, x, y))
        constr[var2].append((var1, y, x))
    return constr
```

```
def initGrid():
    grid = [ ['1A', '', '2D', '', '3D'], ['-', '-', '', '-', ''],
             ['-', '4A', '', '5D', ''], ['6D', '-', '7A', '', ''],
             ['8A', '', '', '', ''], [' ', '-', '-', '-', '']]
    return grid
```

```
# ***** DISPLAY HELPER FUNCTIONS
***** #
```

```
def displayGrid(grid):
    for row in range(6):
        print("\n", "----" * 9, "\n|", end = " ")
        for col in range(5):
            print('{0: <5}|'.format(grid[row][col]), end= " ")
```

```
def organizeResult(domain, position):
```

```

res = [['-' for col in range(5)] for row in range(6)]
for var, domain in domain.items():
    row, col, direction = position[var]
    domain = list(domain[0])
    length, index = len(domain), 0
    while(index < length):
        res[row][col] = domain[index]
        index += 1
        if(direction == "D"):
            row += 1
        else:
            col += 1
return res

```

```

def displayDomain(domain):
    for key, val in domain.items():
        print(key, " : ", val)
    print()

```

```

# ***** CONSTRAINT SATISFACTION ALGORITHM
***** #

```

```

def arc_reduce(li_1, li_2, ind1, ind2):
    change = False
    for ele_1 in li_1:
        flag = True
        for ele_2 in li_2:
            if(list(ele_1)[ind1-1] == list(ele_2)[ind2-1]):
                flag = False
                break
        if(flag == True):
            change = True
            li_1.remove(ele_1)
    return change

```

```

def apply_constraints(domain, constr):
    no_change = True
    for xi in constr.keys():
        for xj, x, y in constr[xi]:
            if(arc_reduce(domain[xi], domain[xj], x, y)):
                if(len(domain[xi]) == 0):
                    return -1
            else:
                no_change = False

```

```
return no_change
```

```
# ***** MAIN FUNCTION  
***** #
```

```
if __name__ == "__main__":
```

```
    domain, position = initDomains()  
    constr = initConstraints()  
    grid = initGrid()
```

```
    print("\n\nGiven question : \n")  
    displayGrid(grid)
```

```
    print("\n\nDomain before optimizations : \n")  
    displayDomain(domain)
```

```
    while(True):  
        no_change = apply_constraints(domain, constr)  
        if(no_change == -1):  
            print("Not possible to solve")  
        elif(no_change == True):  
            break  
        elif(no_change == False):  
            continue
```

```
    print("\n\nReduced domain after applying constraints : \n")  
    displayDomain(domain)
```

```
    print("\n\nFinal Answer : \n")  
    result = organizeResult(domain, position)  
    displayGrid(result)  
    print("\n\n")
```

Output:

```
In [1]: runfile('D:/Semester 6/Artificial Intelligence Lab/Crossword_Puzzle_8.py', wdir='D:/Semester 6/Artificial Intelligence Lab')
```

Given question :

```
-----
| 1A |   | 2D |   | 3D |
|-----|
| -  | -  |   | -  | |
|---|---|---|---|---|
| -  | 4A |   | 5D |   |
|-----|
| 6D | -  | 7A |   |   |
|-----|
| 8A |   |   |   |   |
|-----|
|   | -  | -  |   | -  |
|-----|
```

Domain before optimizations :

```
1A : ['HOSES', 'LASER', 'SAILS', 'SHEET', 'STEER']
4A : ['HEEL', 'HIKE', 'KEEL', 'KNOT', 'LINE']
7A : ['AFT', 'ALE', 'EEL', 'LEE', 'TIE']
8A : ['HOSES', 'LASER', 'SAILS', 'SHEET', 'STEER']
2D : ['HOSES', 'LASER', 'SAILS', 'SHEET', 'STEER']
3D : ['HOSES', 'LASER', 'SAILS', 'SHEET', 'STEER']
5D : ['HEEL', 'HIKE', 'KEEL', 'KNOT', 'LINE']
6D : ['AFT', 'ALE', 'EEL', 'LEE', 'TIE']
```

Reduced domain after applying constraints :

```
1A : ['HOSES']
4A : ['HIKE']
7A : ['LEE']
8A : ['LASER']
2D : ['SAILS']
3D : ['STEER']
5D : ['KEEL']
6D : ['ALE']
```

Final Answer :

```
-----
| H | O | S | E | S |
|-----|
| - | - | A | - | T |
|-----|
| - | H | I | K | E |
|-----|
| A | - | L | E | E |
|-----|
| L | A | S | E | R |
|-----|
| E | - | - | L | - |
|-----|
```

Result:

Thus, Crossword Puzzle has been solved as constraint satisfaction problem by giving set of domains, values and constraints as input.